

⑦

66-5-99

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

昭62-192849

⑬ Int. Cl.

識別記号

庁内整理番号

⑭ 公開 昭和62年(1987)8月24日

G 06 F 15/16
3/06

3 0 1

J-2116-5B
6711-5B

審査請求 未請求 発明の数 1 (全5頁)

⑮ 発明の名称 計算機ネットワークにおけるコマンド実行方式

⑯ 特 願 昭61-34900

⑰ 出 願 昭61(1986)2月19日

⑱ 発 明 者	木 本	隆	川崎市中原区上小田中1015番地	富士通株式会社内
⑱ 発 明 者	中 村	芳 弘	川崎市中原区上小田中1015番地	富士通株式会社内
⑱ 発 明 者	佐 藤	恵 司	川崎市中原区上小田中1015番地	富士通株式会社内
⑱ 発 明 者	深 津	貞 雄	川崎市中原区上小田中1015番地	富士通株式会社内
⑱ 発 明 者	渡 部	信 雄	川崎市中原区上小田中1015番地	富士通株式会社内
⑰ 出 願 人	富士通株式会社		川崎市中原区上小田中1015番地	
⑰ 代 理 人	弁理士 山谷 皓 榮			

明 細 書

1. 発明の名称 計算機ネットワークにおける
コマンド実行方式

2. 特許請求の範囲

複数のコンピュータをネットワークで接続するとともにそのうちの少くとも1つのコンピュータに外部記憶装置を接続してこれをあたかも自コンピュータの外部記憶装置であるように使用することのできるシステムにおいて、

複数のコマンドにより構成されるとともに各コマンドのプログラムが記入されたコマンドファイルが格納された外部記憶手段(Fo)と、

コマンドファイルのコマンド名やバイト数を識別する識別手段(3、3')を具備する複数のコンピュータ(Bo、Co)を有し、

コンピュータで前記コマンドファイルを実行するときに前記外部記憶手段(Fo)よりそのコマンドファイルのデータを連続的に伝送させてこれ

をコマンドファイルとコマンドに区分し、コンピュータでこの区分したコマンドを実行するようにしたことを特徴とする計算機ネットワークにおけるコマンド実行方式。

3. 発明の詳細な説明

(目次)

概要

産業上の利用分野

従来技術

発明が解決しようとする問題点

問題点を解決するための手段

作用

実施例

発明の効果

(概要)

本発明は、複数のコンピュータをネットワークで接続してその1台に外部記憶装置を接続しこれを他のコンピュータがあたかも自コンピュータの

外部記憶であるかの如く使用できる機能を有するシステムにおいて、前記外部記憶装置に予めコマンドファイルとその中で使用するコマンドすべてを1つのファイルにまとめておき、コマンドファイルを実行するコンピュータは実行に先立ち前記ファイルを読み、これをコマンドファイルとコマンドに分解し、自主記憶上に登録し、それからこれを実行するものである。

(産業上の利用分野)

本発明は、計算機ネットワークにおけるコマンド実行方式に係り、特にファイルサーバを有するコンピュータ網において、コマンドファイルを実行するとき、必要なファイルを一体化したものをファイルサーバ上に用意しておき、実行側のコンピュータはこれを自主記憶上に展開することにより実行するようにしたものに関する。

(従来の技術)

コンピュータでは主記憶の容量不足を補うため

に外部記憶装置が使用されている。ところで、第4図に示す如く、複数のコンピュータA、B、C……をLAN（ローカルエリア・ネットワーク）の如き通信路NTを介して接続したコンピュータ・ネットワークでは、個々のコンピュータに外部記憶装置を接続するよりも、例えばコンピュータAに大容量の外部記憶装置（磁気ディスク、DASD等）Fを接続してファイルサーバを構成し、コンピュータB、C……はこの外部記憶装置Fをあたかも自コンピュータの外部記憶であるかの如く使用する機能を具備している。

ところでコンピュータがデータ処理を行う場合、使用者がしばしば行うようなある特定の処理をコマンド列としてファイルに登録し、簡単に実行できるようにしている。この登録されたファイルはコマンドファイルと呼ばれている。

第5図に、このコマンドファイルCOMFILEの1例を示す。これは、コマンドCOM1、COM2、COM3により構成されたものであってコマンドファイル名(FN)が付与されている。

そして、第4図に示すシステムでは、このコマンドファイルは外部記憶装置Fに格納されている。

ところで、外部記憶装置Fに格納されている、第5図に示すコマンドファイルを、例えばコンピュータBが実行する場合、第6図に示す如き動作シーケンスとなる。

① コンピュータBにおいて使用者がそのコマンドファイルのファイル名(FN)をキーインする。これによりコンピュータBよりファイルサーバに対してコマンドファイルに対するOPEN要求が行われる。ファイルサーバはこのコマンドファイル(FN)に対するOPEN処理を行い、COMFILEの内容(データ①)をコンピュータBに転送する。

② コンピュータBではこの転送されたCOMFILEの内容よりこのコマンドファイルがコマンドCOM1、COM2、COM3により構成されていることを認識する。そしてまずコマンドCOM1に対してOPEN要求を行う。ファイルサーバではこれによりOPEN処理を行い、またC

OM1の内容つまりCOM1のプログラムをデータ②としてコンピュータBに転送する。コンピュータBではこれをその主記憶上に展開してこのCOM1のプログラムを実行する。そしてそのあとCOM1のファイルに対するCLOSE要求を行ない、ファイルサーバはこのCOM1に対するCLOSE処理を行う。

③ ところでコンピュータBは前記COM1のCLOSE要求に続いてコマンドCOM2のファイルに対するOPEN要求を行う。これに対してファイルサーバでは前記COM1に対すると同様な処理が行われる。このようなことがCOM3に対しても遂行される。そしてCOM3が実行された後にコンピュータBからCOM3に対するCLOSE要求が送出され、その後COMFILEに対するCLOSE要求が行われてCOMFILEがCLOSE処理される。このようにしてCOMFILE(FN)に対する処理が終了する。

〔発明が解決しようとする問題点〕

このような従来の方式においては、コマンドファイルおよびそのコマンドの個数だけのOPEN、CLOSE処理が必要となる。ところでこのOPEN処理は外部記憶装置Fにおいて必要とするデータをアクセスしたり、外部記憶装置Fから読み出したデータを一時保持するため主記憶の領域を確保したりすることになるので、その処理時間が長く、このようなOPEN、CLOSE処理数が増加するとデータ処理速度がおそくなる。

また第6図に示す動作シーケンスを遂行するため、コマンドファイルに対するOPEN、CLOSEのバケットと、各コマンドファイルに対するバケットに対するバケットが必要になる。コマンドファイル、コマンド転送の際に、許容された最大長のバケットが使用できないことがあり、このためバケット数が増える。第6図において、データ④として示すコマンドファイル転送のデータバケット長は、コマンドファイルの内容が小さいと考えられるので、最大長とはならない。また、第

6図においてデータ④として示すCOM1のプログラムのデータバケットは、COM1の内容であるプログラム転送の際に最大長のバケットを使用したとしても、その最終のデータは通常バケットの最大長ではない。このようにデータ転送の際に少なくとも必要となる最小バケット数(次式)
 最小バケット数 = (コマンドファイルのバイト数 + コマンドのバイト数の合計) ÷ (最大バケット長)

よりも、バケットが多くなる。このようにバケット数が増加すると、コンピュータとして処理速度の遅いもの(パソコン等)を使用した場合などは、バケット処理のオーバーヘッドのために、非常に転送効率が低下し、コマンドファイルの実行が遅くなってしまう。

本発明の目的は、前記の問題点を解決するため、OPEN回数も少なく、バケット数も少ない計算機ネットワークにおけるコマンド実行方式を提供することである。

〔問題点を解決するための手段〕

前記目的を達成するため、本発明では、第1図に示す如く、コマンドファイルと必要なすべてのコマンドを一体化した新しいコマンドファイルをファイルサーバ上に作成しておく。

コンピュータの使用者がこのコマンドファイルの実行を指示すると、このコマンドファイルをファイルサーバから読取り、主記憶上にコマンドファイル、コマンドと分解してロードする。以後、コンピュータはファイルサーバを使用せず主記憶でコマンドファイルを処理する。

〔作用〕

コマンドファイルに対するOPEN要求を行うと、これに対するデータ転送を、第1図のCOMFILEのバイト数からCOM3の内容まで連続的に転送するので、各コマンドCOM1、COM2、COM3毎のOPEN要求は不必要となる。しかも連続的に転送されるので、バケット数を少なくすることができる。

〔実施例〕

本発明の一実施例を第2図、第3図にもとづき他図を参照して説明する。

第2図は本発明の一実施例構成図、第3図は本発明における動作シーケンスを示す。

第2図において、コンピュータBはプロセッサ1および主記憶2より構成され、識別部3を具備している。またコンピュータCはプロセッサ1'および主記憶2'および仮想的に外部記憶とみなした主記憶(通常RAM Diskと呼ばれる)4'より構成され、識別部3'を具備している。

ここで識別部3、3'はファイルサーバから伝達された、第1図に示す如き、コマンドファイルの区切り部分を識別するものであって、COMFILEの区分のバイト数、COM1のバイト数、COM2のバイト数、COM3のバイト数……を認識してそれぞれの区分で区別するものである。

ファイルサーバを構成する外部記憶Fは、第

4図の外部記憶Fに対応するものであるが、第5図に示す如くコマンドCOM1~COM3により構成されるコマンドファイルが格納されるものではなく、第1図に示す如く、COM1、COM2、COM3については、それらを構成する具体的なプログラムが記入されたコマンドファイルとして格納されている。本発明におけるコマンドファイルは、そのコマンドファイルの構成を示すCOMFILE部分は、従来のものと同様に、そのCOMFILE部分のバイト数と、COMFILEの内容(この例ではCOM1~COM3より構成されていることの指示)が記入されているが、それに続いて、COM1のバイト数、ファイル名記入部およびCOM1の内容であるプログラム記入部、COM2のバイト数、ファイル名記入部およびCOM2のプログラム記入部、COM3のバイト数、ファイル名記入部およびCOM3のプログラム記入部が存在する。この新しいコマンドファイルはコマンドファイル生成時に、ファイルサーバ上に作製されるものである。

てこれらをもとどおりに区分して、その主記憶2上にロードする。

④ コンピュータB0ではこのようにして主記憶2上に登録されたコマンドファイルを解釈して、主記憶2上に登録されているコマンドCOM1~COM3のプログラムを使用してこれらを順次実行する。そして実行終了後、プロセッサ1は、ファイルサーバに対しそのコマンドファイルに対するCLOSE要求が行われ、ファイルサーバではこれによりこのコマンドファイルのCLOSE処理を行う。

前記④においてももとのファイルのバイト数と、ファイル名を付加してあるので、これを受信したコンピュータ側では主記憶上にコマンドファイルやコマンドに分解しなおすことができる。

ところで、コンピュータのオペレーティング・システムによっては、主記憶上にコマンド等のロードができない場合がある。第2図におけるコンピュータC0のオペレーティング・システムがこのようなものであるとき、そのオペレーティング・シ

いま、コンピュータB0の使用者が、そのコマンドファイル名(FN)をキーインしてその実行を指示すると、第3図に示す如きシーケンスが遂行される。

① 前記コマンドファイル名をキーインしその実行を指示すると、プロセッサ1よりファイルサーバに対してそのコマンドファイルに対するOPEN要求が行われる。ファイルサーバはこのコマンドファイル(FN)に対するOPEN処理を行ない、それからこのコマンドファイルをデークとしてコンピュータB0に転送する。

② この転送デークは、識別部3によりまずそのCOMFILEのバイト数とCOMFILEの内容から、コマンドがCOM1~COM3よりなるものであることを認識する。そして次にCOM1のバイト数、ファイル名を検出しCOM1の内容の終わりを検出する。以下同様にしてCOM2のバイト数、ファイル名、COM3のバイト数、ファイル名を検出する。このようにしてコマンドファイルとコマンドCOM1~COM3に識別し

テムに主記憶の一部を仮想的に外部記憶4'とみなし仮想的に使用する機能を付加する。そして一般にRAMDISKと呼ばれるこの仮想的な外部記憶上に分解したファイルを置き、この上で実行することになる。

本発明では、予め、第1図に示す如き、新しいコマンドファイルを作り出さなければならないので、例えばIPL時のSTARTUPFILEの実行とか、コンパイルを行うときのコマンドファイルの実行のように、頻繁に行われるコマンドファイルの実行に適している。

(発明の効果)

本発明によればファイルサーバでのファイルのOPEN、CLOSEが各1回でよいので、ファイルサーバのファイルアクセスの負荷を大幅に減少させることができる。

また、コマンドファイルをコマンド毎に区分せずに連続的に転送するため、パケットの最大転送長を最大限度利用することができるため、パケッ

ト数が最低限必要な数とほぼ同じになるため、従来の場合に比較して、パケットの処理のオーバーヘッドを非常に少なくできる。

4. 図面の簡単な説明

第1図は本発明の原理説明図であって本発明において使用されるコマンドファイル例、

第2図は本発明の一実施例構成図、

第3図は本発明における動作シーケンス、

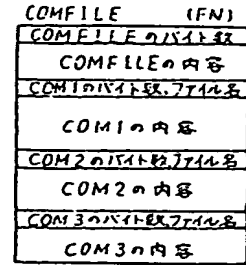
第4図はコンピュータ・ネットワーク例、

第5図はコマンドファイル例、

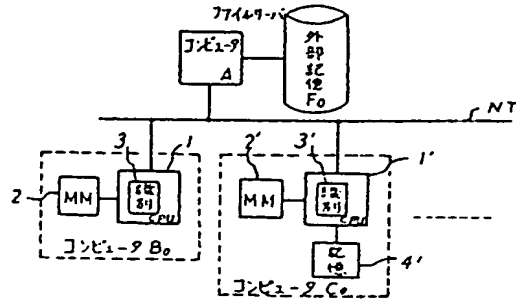
第6図はコマンドファイルを実行するときの動作シーケンスである。

- 1、1'プロセッサ 2、2'主記憶
3、3'識別部

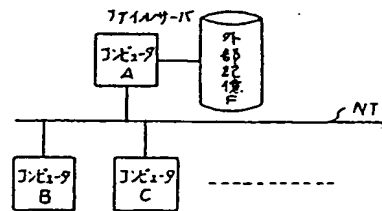
特許出願人 富士通株式会社
代理人弁理士 山 谷 晴 榮



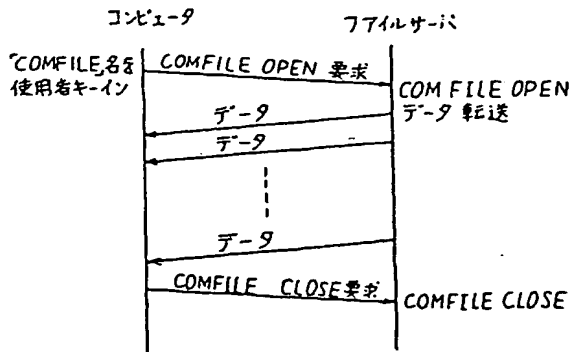
本発明の原理説明図
第1図



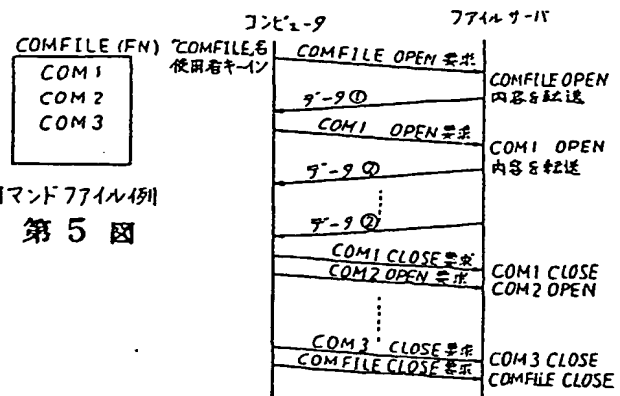
本発明の一実施例構成
第2図



コンピュータネットワーク
第4図



本発明による動作シーケンス
第3図



コマンドファイル例
第5図

コマンドファイルを実行するときの
動作シーケンス
第6図

Translator's Notes for Laid-Open No. SHO 62-192849:

5 In line 4 in claim, "all the computers" is a translation,
based on educated guess, of a term led by an illegibly printed
character that appears in line 3 in the claim in the original
Japanese document.

10 Please check the illegible character preceding "コン
ピュータ (computer)" in line 3 in the Japanese claim to verify
the translation.

15 Considering similar phrases appearing in other
locations, the originally printed term may be an equivalent
of "the computer itself", which would be assumed to be a
typographical error.

20 In line 8, page 6, "the final data" is a translation
of illegibly printed characters preceding "のデータ (data)"
appearing in line 4, column 2, page 299 of the original
document. Please check it or send me a clear copy for correct
translation.

Japanese Patent Application Laid-Open No. SHO 62-192849

Laid-Open Date: August 24, 1987

Application No. SHO 61-34900

Application Date: February 19, 1986

5 Applicant: Fujitsu KK

Inventors: KIMOTO Takashi, NAKAMURA Yoshihiro,

SATOU Tadashi, FUKAZU Tadao, and WATABE Nobuo

Agent: YAMATANI Kouei

SPECIFICATION

10

1. TITLE OF THE INVENTION

COMMAND EXECUTION SYSTEM IN COMPUTER NETWORK

2. CLAIM

15

In a system in which a plurality of computers are connected via a network, and an external storage device is connected to at least one of the computers, and the external storage device is allowed to be used as if it were the external storage device of all the computers, a command execution

20

system in a computer network being characterized in:

that the command execution system comprises:

external storage means (F_0) in which a command file that includes a plurality of commands and includes a program of each command written in is stored; and

25

a plurality of computers (B_0, C_0) having identification means ($3, 3'$) for identifying a command name and a byte count of the command file, and

that when the command file is to be executed by a computer, the data of the command file is consecutively transferred from the external storage device (F₀), and separated into the command file and commands, and the
5 separated commands are executed by the computer.

3. DETAILED DESCRIPTION OF THE INVENTION

[INDEX]

SUMMARY

10 FIELD OF INDUSTRIAL APPLICATION

PROBLEMS TO BE SOLVED BY THE INVENTION

MEANS FOR SOLVING THE PROBLEMS

OPERATION

EMBODIMENT

15 ADVANTAGES OF THE INVENTION

[SUMMARY]

According to the invention, in a system in which a plurality of computers are connected via a network, and in
20 which an external storage device is connected to one of the computers, and in which the other computers are allowed to use the external storage device if it were the external storage device of those computers, the external storage device pre-stores a single file combining a command file and
25 all the commands used therein. A computer that is about to execute the command file reads the single file and disintegrates the file into the command file and the commands

before the execution. After the computer registers the command file and the commands at its own main memory, the computer executes the command file.

5 [FIELD OF INDUSTRIAL APPLICATION]

The present invention relates to a command execution system in a computer network and, more particularly, to a system in a computer network including a file server, wherein when a command file is to be executed, a computer that executes
10 the command file develops a unified file of necessary files prepared in the file server, onto the main memory of the computer, in order to execute the command file.

[CONVENTIONAL ART]

15 For a computer, an external storage device is normally used to supplement an insufficient capacity of the main memory of the computer. In a computer network as shown in Fig. 4, where a plurality of computers A, B, C... are connected via a communication path NT as in a LAN (local area network),
20 a large-capacity external storage device (a magnetic disk, a DASD, or the like) F is connected to, for example, the computer A to form a file server, instead of connecting external storage devices separately to the individual computers. Each of the computers B, C... is equipped with
25 the function to use the external storage device F as if the external storage device F were an external storage device of its own.

For computer data processing, a specific processing, for example, a processing frequently conducted by a user, is registered as a command sequence in a file so that the processing can easily be performed. Such a registered file
5 is termed command file.

Fig. 5 exemplifies a command file COMFILE. The command file COMFILE comprises commands COM1, COM2, COM3, and is assigned with a command file name (FN). In the system shown in Fig. 4, the command file is stored in the external storage
10 device F.

If the command file, as shown in Fig. 5, which is stored in the external storage device F, is executed by, for example, the computer B, an operation sequence is performed as shown in Fig. 6.

15 (1) A user keys the file name (FN) of the command file into the computer B, so that the computer B sends an OPEN request regarding the command file to the file server. The file server performs an OPEN processing for the command file (FN), and transfers the content of COMFILE (data (1))
20 to the computer B.

(2) Based on the transferred content of COMFILE, the computer B recognizes that the command file comprises the commands COM1, COM2, COM3. Then, the computer first makes an OPEN request regarding the command COM1. The file server
25 correspondingly performs an OPEN processing, and transfers to the computer B the content of COM1, that is, the program of COM1, as data (2). The computer B develops the data on

its main memory, and executes the program of COM1.

Subsequently, the computer B makes a CLOSE request regarding the COM1 file, and the file server performs a CLOSE processing for the COM1.

5 (3) Subsequently to the CLOSE request regarding COM1, the computer B makes an OPEN request regarding the COM2 file. In response, the file server performs a processing similar to that performed for COM1. A similar operation is also performed for COM3. After execution of COM3, the computer
10 B sends a CLOSE request regarding COM3, and then sends a CLOSE request regarding COMFILE, so that the CLOSE processing for COMFILE is performed. Thus, the processing for COMFILE (FN) is completed.

15 [PROBLEMS TO BE SOLVED BY THE INVENTION]

The above-described conventional system needs a number of OPEN and CLOSE processings equal to the number of command files and commands. An OPEN processing requires a long processing time since, for example, necessary data in the
20 external storage device F needs to be accessed and an area of the main memory needs to be secured in order to temporarily store data read from the external storage device F. Therefore, an increase in the number of OPEN and CLOSE processings slows the data processing rate.

25 Furthermore, in order to perform the operation sequence shown in Fig. 5, packets for OPEN and CLOSE of the command file and packets for the individual command file

packets are needed. There are cases at the time of transferring the command file or a command where a packet of a maximum allowable length cannot be used and, therefore, the number of packets used increases. In Fig. 6, the data packet length for transfer of a command file indicated as "DATA (1)" is not the maximum length since the content size of the command file is considered small. As for the data packet of the program of COM1 indicated as "DATA (2)" in Fig. 6, the final data normally does not become the maximum packet length even if a packet of the maximum length is used to transfer the content of COM1, that is, the program. In this manner, the number of packets actually used for data transfer becomes greater than the minimum number of packets needed for data transfer (the following equation).

Minimum number of packets = (Byte count of command file + the total byte count of commands) / (Maximum packet length)

A thus-increased number of packets inconveniently reduces the transfer efficiency, for example, in a case where a low processing speed computer (a personal computer or the like) is used, due to the overhead of the packet processing. Therefore, the execution of the command file is delayed.

It is an object of the invention to provide a command execution system in a computer network wherein the number of times of executing an OPEN processing is small, and the number of packets used is small, in order to solve the above-stated problems.

[MEANS FOR SOLVING THE PROBLEMS]

To achieve the aforementioned object, the invention employs a construction where a new command file combining a command file and all the necessary commands as indicated
5 in Fig. 1 is prepared in a file server.

Upon an instruction made by a computer user to execute the command file, a computer reads the command file from the file server, and disintegrates it into the command file and commands and loads them into the main memory of the computer.
10 After that, the computer processes the command file using its main memory, without using the file server.

[OPERATION]

Upon an OPEN request regarding the command file, the
15 corresponding data, covering from the byte count of COMFILE and the content of COM3 indicated in Fig. 1, is consecutively transferred. Therefore, the OPEN requests regarding the commands COM1, COM2, COM3 become unnecessary. Furthermore, since the transfer is consecutively performed, the number
20 of packets used can be reduced.

[EMBODIMENT]

An embodiment of the present invention will be described with reference mainly to Figs. 2 and 3 and also
25 to the other drawings.

Fig. 2 illustrates the construction of the embodiment of the present invention. Fig. 3 illustrates an operation

sequence according to the invention.

Referring to Fig. 2, a computer B₀ comprises a processor 1 and a main memory 2, and also has an identifier portion 3. A computer C₀ comprises a processor 1', a main memory 2',
5 and a main memory (normally termed RAM disk) 4' virtually regarded as an external storage, and also has an identifier portion 3'.

Each identifier portion 3, 3' identifies partition portions of a command file, as shown in Fig. 1, which is
10 transmitted from a file server. Each identifier portion recognizes the byte count of a segment of COMFILE, the byte count of COM1, the byte count of COM2, the byte count of COM3..., and discriminates them based on the individual segments.

15 An external storage device F₀, constituting the file server, corresponds to the external storage device F shown in Fig. 4. However, the external storage device F₀ does not store a command file formed by commands COM1-COM3 as indicated in Fig. 5, but stores a command file as indicated in Fig.
20 1, in which specific programs constituting COM1, COM2 and COM3, respectively, are written. In the command file according to the invention, the COMFILE portion indicating the structure of the command file includes the byte count of the COMFILE portion and the content of the COMFILE (in
25 this embodiment, indication of the structure formed by COM1-COM3) that are written, as in the conventional art. However, in the command file according to the invention, the

COMFILE portion is followed by a portion in which the byte count and the file name of COM1 are written, and a portion in which the program of COM1, that is, the content thereof, is written, and a portion in which the byte count and the file name of COM2 are written, and a portion in which the program of COM2 is written, and a portion in which the byte count and the file name of COM3 are written, and a portion in which the program of COM3 is written. This new command file is created in the file server at the time of creating a command file.

When a user of the computer B_0 keys in the command file name (FN) to instruct the computer B_0 to execute the command file, a sequence as indicated in Fig. 3 is performed.

(1) Upon an instruction to execute the command file made by keying in the command file name, the processor 1 outputs an OPEN request regarding the command file to the file server. The file server performs an OPEN processing for the command file (FN), and transfers the command file as data to the computer B_0 .

(2) Based on the byte count of COMFILE and the content of COMFILE obtained from the transferred data, the identifier portion 3 recognizes that the commands comprise COM1-COM3. Subsequently, the identifier portion 3 detects the byte count and the file name of COM1, and detects the end of the content of COM1. Similarly, the identifier portion 3 detects the byte count and the file name of COM2, and the byte count and the file name of COM3. The command

file and the commands COM1-COM3 are thus identified and then separated as they were before, and then loaded into the main memory 2.

(3) The computer B₀ interprets the command file registered at the main memory 2 as described above, and serially executes the commands by using the programs of the commands COM1-COM3 registered in the main memory 2. After execution of the commands, the processor 1 outputs a CLOSE request regarding the command file to the file server. In response, the file server performs a CLOSE processing for the command file.

Since the byte counts and the names of the original files are prepared in the command file in advance, the computer, upon receiving the data, can disintegrate it back into the command file and the commands in the main memory, as described in paragraph (2).

There may be a case where the loading of commands and the like into the main memory is impossible, depending on the operating system of a computer. If the computer C₀ shown in Fig. 2 has such an operating system, the operating system is provided with an additional function to virtually regard and virtually use a portion of the main memory as the external storage 4'. The disintegrated files are placed and executed in the virtual external storage, which is normally termed RAM disk.

According to this invention, a new command file as indicated in Fig. 1 needs to be created in advance. Therefore,

the invention suitable for execution of a command file that is frequently conducted, for example, execution of a start-up file at the time of IPL, execution of a command file for compilation, and the like.

5

[ADVANTAGES OF THE INVENTION]

According to the present invention, since each of the file OPEN processing and the file CLOSE processing needs only to be performed once in the file server, the file access load
10 of the file server can be considerably reduced.

Furthermore, since a command file is consecutively transferred without separating it into commands, the maximum packet length for transfer can be fully utilized, so that the number of packets used becomes substantially equal to
15 the minimum number of packets needed. Therefore, the overhead of the packet processing can be drastically reduced, in comparison with the conventional art.

4. BRIEF DESCRIPTION OF THE DRAWINGS

20 Fig. 1 is an illustration of the principle of the invention, exemplifying a command file used in the invention;

Fig. 2 illustrates the construction of the embodiment of the invention;

Fig. 3 illustrates an operation sequence according to
25 the invention;

Fig. 4 exemplifies a computer network;

Fig. 5 exemplifies a command file; and

Fig. 6 illustrates an operation sequence to execute the command file.

- 1, 1' ... processor
- 5 2, 2' ... main memory
- 3, 3' ... identifier portion

Applicant: Fujitsu KK

Agent: Patent Attorney YAMATANI Kouei

10

FIG. 1

ILLUSTRATION OF PRINCIPLE OF INVENTION

BYTE COUNT OF COMFILE

5 CONTENT OF COMFILE

 BYTE COUNT, FILE NAME OF COM1

 CONTENT OF COM1

 BYTE COUNT, FILE NAME OF COM2

 CONTENT OF COM2

10 BYTE COUNT, FILE NAME OF COM3

 CONTENT OF COM3

FIG. 2

CONSTRUCTION OF EMBODIMENT OF INVENTION

15

 FILE SERVER

 EXTERNAL STORAGE DEVICE F_0

 COMPUTER A

20

 COMPUTER B_0

 3. IDENTIFIER

 COMPUTER C_0

25 3'. IDENTIFIER

 4'. STORAGE

FIG. 3

OPERATION SEQUENCE ACCORDING TO INVENTION

1. COMPUTER
- 5 2. COMFILE NAME KEYED IN BY USER
3. COMFILE OPEN REQUEST
4. DATA
5. DATA
6. DATA
- 10 7. COMFILE CLOSE REQUEST
8. FILE SERVER
9. TRANSFER DATA

FIG. 5

- 15 EXAMPLE OF COMMAND FILE

FIG. 6

OPERATION SEQUENCE TO EXECUTE COMMAND FILE

- 20 A. COMPUTER
- B. COMFILE NAME KEYED IN BY USER
- C. COMFILE OPEN REQUEST
- DATA (1)
- D. COM1 OPEN REQUEST
- 25 DATA (2)
- DATA (3)
- E. COM1 CLOSE REQUEST

- F. COM2 OPEN REQUEST
- G. COM3 CLOSE REQUEST
- H. COMFILE CLOSE REQUEST
- I. FILE SERVER
- 5 J. TRANSFER CONTENT
- K. TRANSFER CONTENT

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.